Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 576–583
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Incremental extreme learning machine with fully complex hidden nodes

Guang-Bin Huang[a,*], Ming-Bin Li[a], Lei Chen[b], Chee-Kheong Siew[a]

[a]School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore
[b]School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore

Available online 1 October 2007

Abstract

Huang et al. [Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17(4) (2006) 879–892] has recently proposed an incremental extreme learning machine (I-ELM), which randomly adds hidden nodes incrementally and analytically determines the output weights. Although hidden nodes are generated randomly, the network constructed by I-ELM remains as a universal approximator. This paper extends I-ELM from the real domain to the complex domain. We show that, as long as the hidden layer activation function is complex continuous discriminatory or complex bounded nonlinear piecewise continuous, I-ELM can still approximate any target functions in the complex domain. The universal capability of the I-ELM in the complex domain is further verified by two function approximations and one channel equalization problems.

Keywords: Feedforward networks; Complex activation function; Constructive networks; ELM; I-ELM; Channel equalization

1. Introduction

*S*ingle-hidden *l*ayer *f*eedforward neural *n*etworks (SLFNs) have attracted extensive interest in many research and application fields due to their approximation capability [3,24,27]. According to the conventional learning theories [3,24,27], all hidden node parameters (the input weights $\mathbf{a}_i$ of the connections linking the input layer to the hidden layer and the biases b_i of the additive hidden nodes, or the centers $\mathbf{a}_i$ and the impact factors b_i of the RBF hidden nodes) need to be tuned in order to make SLFNs work as universal approximators. Several researchers [4,9,18,26] have independently found that the input weights or centers $\mathbf{a}_i$ need not be tuned.

(1) Baum [4] has claimed that (seen from simulations) one may fix the weights of the connections on one level and simply adjust the connections on the other level and no (significant) gain is possible by using an algorithm able to adjust the weights on both levels simultaneously. Baum [4] did not discuss whether all the hidden node biases b_i should be set with the same value. Baum [4] did not discuss either whether the hidden node biases b_i should be tuned or not.

(2) Lowe [26] found that from an interpolation (instead of universal approximation) point of view the centers $\mathbf{a}_i$ of RBF hidden nodes can be randomly selected from the training data instead of tuning. In Lowe's learning model, the impact factor b_i of RBF hidden nodes is not randomly selected and it depends on the spread of the training data sets. Furthermore all the impact factors b_i are usually set with the same value [26, p. 173]. Seen from Broomhead and Lowe [5], Lowe et al. [5,26] in fact focuses on a specific RBF network with the same impact factor b assigned to all the RBF hidden nodes: $f_n(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g(b\|\mathbf{x} - \mathbf{a}_i\|)$, $\mathbf{x} \in \mathbf{R}^d$ (cf. [5, Eq. (2.2)]). If RBF centers and impact factors are selected based on the training data, it may give advantages to the training data and thus easily causes overfitting. (ELM works on generalized feedforward network [10,11] and RBF hidden node type is just one of the specific case of ELM. Different from the RBF network presented in Lowe et al. [5,26], the main RBF network interested by ELM is $f_n(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g(b_i\|\mathbf{x} - \mathbf{a}_i\|)$ where the RBF hidden nodes are not requested to have the same impact factors b_i.) Interestingly, RBF networks $f_n(\mathbf{x}) \sum_{i=1}^{n} \beta_i g(b\|\mathbf{x} - \mathbf{a}_i\|)$ with randomly generated

*Corresponding author. Tel.: +65 6790 4489; fax: +65 6793 3318.
E-mail address: egbhuang@ntu.edu.sg (G.-B. Huang).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.07.025

centers $\mathbf{a}_i$ and randomly generated same values of impact factors b in fact does not generally have the universal approximation capability, in contrast, RBF networks $f_n(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g(b_i\|\mathbf{x} - \mathbf{a}_i\|)$ with randomly generated centers $\mathbf{a}_i$ and randomly generated impact factors b_i does generally have the universal approximation capability [10,11].

(3) Igelnik and Pao [18] proposed a random vector version of the functional-link (RVFL) net. In RVFL model, the input weights $\mathbf{a}_i$ are "uniformly" drawn from a probabilistic space $V_\alpha^d = [0, \alpha\Omega] \times [-\alpha\Omega, \alpha\Omega]^{d-1}$ (d: the input dimension). The hidden node biases b_i depend on the weights $\mathbf{a}_i$ and some other parameters $\mathbf{y}_i$ and u_i: $b_i = -(\alpha\mathbf{a}_i \cdot \mathbf{y}_i + u_i)$, where $\mathbf{y}_i$ and u_i are randomly generated from $[0, 1]^d$ and $[-2\Omega, 2\Omega]$. α and Ω have to be determined in the learning stage and depends on the training data distribution. However, Igelnik and Pao [18] does not show how to determine α and Ω in the learning stage.

(4) Ferrari and Stengel [9] also found that the input weights $\mathbf{a}_i$ need not be trained, however, similar to Igelnik and Pao [18], Ferrari and Stengel [9] thought that there should have some dependence between the hidden node biases b_i, the weights $\mathbf{a}_i$ and the training data.

Thus, strictly speaking, in all the previous works [4,5,9,18,26] the so-called "randomly" generated hidden node parameters are not completely independent of the training data. For example, the hidden node biases or the impact factors b_i can only be generated after seeing the training data. In this sense, these works still belong to the conventional tuning-based learning models where the hidden node parameters are generated only after the training data are presented.

Different from [4,5,9,18,26], according to the best of our knowledge Tamura and Tateishi [28] first proves that from the interpolation (instead of universal approximation) point of view SLFNs with an infinite differentiable sigmoid activation function and with both randomly generated input weights $\mathbf{a}_i$ and hidden node biases b_i can approximate the training data with arbitrarily small errors. In Tamura and Tateishi's model [28] both the input weights $\mathbf{a}_i$ and hidden node biases b_i can be randomly generated fully independently from the training data. There is no necessary relationship between the input weights $\mathbf{a}_i$ and the hidden node biases b_i either. However, for SLFN cases, in order to learn N distinct training data N hidden nodes are required, which leads to overfitting and may not work well in practical applications. Furthermore, generally speaking, Tamura and Tateishi's model [28] does not have the universal approximation capability which is required by all the function approximators.

Based on these earlier works, recently, Huang et al. [10,12–17,25] have proposed a series of novel learning methods called extreme learning machines (ELM) for different applications. Different from the above-mentioned semi-tuning-based learning methods [4,5,9,18,26] which, strictly speaking, only randomly select the input weights or centers $\mathbf{a}_i$ instead of all parameters of the hidden nodes, ELM is fully automatically implemented and in theory no intervention is required from users, all the hidden node parameters $\mathbf{a}_i$ and b_i are randomly generated independently of the target functions and the training patterns. We found that from the function approximation point of view there is no relationship between $\mathbf{a}_i$ and b_i and the hidden node parameters can be irrelevant to the target functions and the training data. The output layer weights can then be analytically determined by using a least-squares method. Since ELM does not adjust hidden node parameters and need not find the relationship between the input weights (or RBF centers) $\mathbf{a}_i$ and the hidden node bias (or impact factors) b_i, ELM is extremely simple and can run extremely fast. Huang et al. [11] has proved the universal approximation capability of ELM in an incremental method (I-ELM). ELM with any bounded nonlinear piecewise continuous activation functions can work as universal approximators. For example, ELM can be used to train SLFNs with a hardlimit type of hidden layer which cannot be handled by all the earlier methods [4,5,9,18,26]. Huang and Chen [10] has recently extended the earlier work [11] to more generalized cases and shows that: if SLFNs (with piecewise continuous computational hidden nodes) can work as universal approximators with adjustable hidden parameters, from the function approximation point of view the hidden node parameters of such "generalized" SLFNs (including sigmoid networks, RBF networks, trigonometric networks, threshold networks, high-order networks, etc.) can actually be randomly generated according to any continuous sampling distribution. Most of these works are focused on the real domain.

Li et al. [25] have extended ELM from the real domain to the complex domain which is referred to as C-ELM, but its universal approximation capability has not been investigated yet. Different from many other complex domain learning algorithms, C-ELM can be applied in SLFNs with fully complex instead of complex-valued activation functions. Although neural networks have been successfully used in complex fields such as wireless and mobile communication applications [6,7,19], it faces the challenge in finding proper nonlinear fully complex activation functions to construct neural networks to process complex signal [20–22]. According to complex analysis, there may exist some conflicts between the boundedness and the differentiability of complex function in the entire complex domain [22]. A bounded analytic (differentiable at every point $z \in C$) function must be a constant in the complex domain C. Recently, Kim and Adali [20] proved the approximation capability of SLFNs with tunable hidden nodes and with fully complex activation functions.

In this paper, we further extend I-ELM into complex domain, we rigorously prove that I-ELM and C-ELM with fully complex activation functions and with randomly generated hidden nodes independent of the training data

can work as universal approximators. More generally, in both I-ELM and C-ELM, the hidden nodes need not be a single additive type, a multiplicative combination of multiple complex additive nodes can be used in the hidden layer.

2. Preliminaries

2.1. Review of I-ELM in real domain

In this section, we first introduce the I-ELM [11] which in the real domain adds randomly generated hidden nodes incrementally. The hidden node parameters $\mathbf{a}_i$ and b_i in I-ELM are not only independent of each other and the training data.

Without any loss of generality, we assume that the network has only one linear output node. All the analysis can be easily extended into multi-nonlinear output nodes cases. A standard SLFNs functions with n hidden nodes can be represented by

$$f_n(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g_i(\mathbf{x}), \quad \mathbf{x} \in \mathbf{R}^d, \ \beta_i \in R, \tag{1}$$

where $g_i(\mathbf{x}) = g(\mathbf{a}_i, b_i, \mathbf{x})$ denotes the output of the ith hidden node: $g_i(\mathbf{x}) = g(\mathbf{a}_i \cdot \mathbf{x} + b_i)$ (for additive nodes) or $g_i(\mathbf{x}) = g(b_i\|\mathbf{x} - \mathbf{a}_i\|)$ (for RBF nodes), β_i is the output weights of the connections linking the ith hidden layer to the output node. I-ELM randomly adds the hidden nodes to the existing networks. The parameters of the hidden nodes $\mathbf{a}_i$ and b_i are randomly generated based on any continuous sampling distribution probability and fully independent of the training data. Once the hidden nodes have been added all the parameters $\mathbf{a}_i$ and b_i of the hidden nodes and their corresponding output weights β_i will be fixed forever.

Let $e_n \equiv f - f_n$ denote the residual error function for the current network f_n with n hidden nodes where $f \in L^2(X)$ is the target function. The mathematical form of I-ELM [11] is

$$f_n(\mathbf{x}) = f_{n-1}(\mathbf{x}) + \beta_n g_n(\mathbf{x}), \tag{2}$$

Unlike other traditional algorithms, which usually find proper parameter based on some optimization techniques such that $\lim_{n\to\infty} \|f - f_n\| = 0$, I-ELM can still work as universal approximators though hidden node parameters are chosen randomly. The corresponding theorem is following:

Lemma 2.1 (*Huang et al. [11]*). *Given any bounded non-constant piecewise continuous function* $g : \mathbf{R} \to \mathbf{R}$ *for additive nodes or any integrable piecewise continuous function* $g : \mathbf{R} \to \mathbf{R}$ *and* $\int_{\mathbf{R}} g(x)\,dx \neq 0$ *for RBF nodes, for any continuous target function* f *and any randomly generated function sequence* $\{g_n\}$, $\lim_{n\to\infty} \|f - f_n\| = 0$ *holds with probability one if*

$$\beta_n = \frac{\langle e_{n-1}, g_n\rangle}{\|g_n\|^2}. \tag{3}$$

Remark 1. I-ELM works with a broad class of activation functions: the activation functions for additive nodes can be any bounded non-constant piecewise continuous functions $g : R \to R$ and the activation functions for RBF nodes can be any integrable piecewise continuous functions $g : R \to R$ and $\int_R g(x)\,dx \neq 0$. I-ELM is not only efficient for SLFNs with continuous (including non-differentiable) activation functions but also for SLFNs with piecewise continuous (such as threshold) activation functions.

2.2. Symbols and theorems in the complex domain

The output of SLFNs with n hidden nodes can be represented by

$$f_n(\mathbf{z}) = \sum_{i=1}^{n} \beta_i g_i(\mathbf{z}), \quad \mathbf{z} \in \mathbf{C}^d, \ \beta_i \in C, \tag{4}$$

where $g_i(\mathbf{z})$ is the output of the ith hidden node for an input vector $\mathbf{z} \in \mathbf{C}^d$.

Let $L^2(Z)$ be a space of functions f with a measurable compact subset Z in the d-dimensional space $\mathbf{C}^d$ such that $|f|^2$ are integrable. For $u, v \in L^2(Z)$, the inner product $\langle u, v\rangle$ is defined by

$$\langle u, v\rangle = \int_Z u(\mathbf{z})\overline{v(\mathbf{z})}\,d\mathbf{z}. \tag{5}$$

The norm in L^2 space will be denoted as $\|\cdot\|$, and the closeness between network function f_n and the target function f is measured by the L^2 distance:

$$\|f_n - f\| = \left[\int_Z (f_n(\mathbf{z}) - f(\mathbf{z}))\overline{(f_n(\mathbf{z}) - f(\mathbf{z}))}\,d\mathbf{z}\right]^{1/2}. \tag{6}$$

In this paper, the sample input space Z is always considered as a bounded measurable compact subset of the space $\mathbf{C}^d$.

Similarly, we define a random function sequence on the complex domain as:

Definition 2.1. The function sequence $\{g_n = g(\mathbf{a}_n \cdot \mathbf{z} + b_n)\}$ is said to be randomly generated if the corresponding parameters $(\mathbf{a}_n, b_n)$ are randomly generated from $\mathbf{C}^d \times C$ based on a continuous sampling distribution probability.

2.3. Necessary lemmas

Some lemmas that are required in the proof of our main theorem are provided in this section.

Lemma 2.2 (*Kolmogorov and Fomin [23, p. 81]*). *The space of* L^2 *is complete.*

Lemma 2.3 (*Kim and Adali [22, Theorem 1]*). *Let* $\sigma : C \to C$ *be any complex continuous discriminatory function. Let* $\mathscr{I}_d$ *denote the d-dimensional complex unit cube* $[0, 1]^d$. *Then the finite sums of the product of the form* $f_n(\mathbf{z}) = \sum_{i=1}^{n} \beta_i \prod_{l=1}^{s_i} \sigma(\mathbf{a}_{il} \cdot \mathbf{z} + b_i)$ *are dense in* $C(\mathscr{I}_d)$, *that is,* $\forall f \in C(\mathscr{I}_d)$ *and* $\varepsilon > 0$, $\exists f_n(\mathbf{z})$ *such that* $|f_n(\mathbf{z}) - f(\mathbf{z})| < \varepsilon$, $\forall \mathbf{z} \in \mathscr{I}_d$, *where* $\mathbf{a}_{il} \in \mathbf{C}^d$ *and* $b_i \in C$.

Lemma 2.3 shows that if σ is complex continuous discriminatory, for any target complex continuous function f there exists f_n such as f_n converges to f everywhere in the bounded set $\mathscr{I}_d$, thus we further have $\|f_n(\mathbf{z})-f(\mathbf{z})\|<\varepsilon$ which is weaker than $|f_n(\mathbf{z})-f(\mathbf{z})|<\varepsilon$. Therefore, we have

Lemma 2.4. *Given any complex continuous discriminatory function $\sigma: C\to C$, for any target continuous function f and $\varepsilon>0$ there exists f_n such that $\|f_n(\mathbf{z})-f(\mathbf{z})\|=\|\sum_{i=1}^{n}\beta_i\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)-f\|<\varepsilon$, where $\mathbf{z}$ is a compact subset of $\mathbf{C}^d$ $\mathbf{a}_{il}\in\mathbf{C}^d$ and $b_i\in C$.*

Lemma 2.5 (*Kim and Adali [22, Theorem 2]*). *Let $\sigma: C\to C$ be any complex bounded measurable discriminatory function. Then the finite sums of the form $f_n(\mathbf{z})=\sum_{i=1}^{n}\beta_i\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)$ are dense in $L^1(\mathscr{I}_d)$, where $\mathbf{a}_{il}\in\mathbf{C}^d$ and $b_i\in C$.*

Lemma 2.4 shows the case where the activation function σ is complex continuous discriminatory, however, in this case the activation function σ may not be bounded. Lemma 2.5 shows the case where the activation function σ is bounded but may not be continuous. As the supremum norm in $L^1(\mu)$ can be generalized to $L^p(\mu)$-norm with $0<p<\infty$, σ may be piecewise continuous, we further have

Lemma 2.6. *Given any complex bounded nonlinear piecewise continuous function $\sigma: C\to C$, $f_n(\mathbf{z})=\sum_{i=1}^{n}\beta_i\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)$ are dense in $L^2(Z)$, where $\mathbf{a}_{il}\in\mathbf{C}^d$ and $b_i\in C$.*

3. Incremental fully complex ELM

3.1. Function approximation

In this subsection we can first show that any continuous target function $f:\mathbf{C}^d\to C$ can be approximated with any arbitrarily small error by an incremental fully complex ELM where the complex hidden nodes are randomly added one by one and will be fixed once added. In fact, given any complex continuous discriminatory or any complex bounded nonlinear piecewise continuous function $\sigma: C\to C$, and any randomly generated function sequence $\{g_i(\mathbf{z})\}$:

$$g_i(\mathbf{z})=\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i), \tag{7}$$

where $\mathbf{a}_{il}$ and b_i are randomly generated fully independently of the target function f based on any continuous distribution probability, then for any small positive value ε, there exists a network sequence $\{f_n\}$, we have $\lim_{n\to\infty}\|f_n-f\|=0$ if $\beta_n=\langle e_{n-1},g_n\rangle/\|g_n\|^2$.

Theorem 3.1. *Given any complex continuous discriminatory or any complex bounded nonlinear piecewise continuous function $\sigma: C\to C$, for any target complex continuous function $f:\mathbf{C}^d\to C$ and any randomly generated function sequence $\{g_n=\prod_{l=1}^{s_n}\sigma(\mathbf{a}_{nl}\cdot\mathbf{z}+b_n)\}$, $\lim_{n\to\infty}\|f-f_n\|=0$ holds with probability one if*

$$\beta_n=\frac{\langle e_{n-1},g_n\rangle}{\|g_n\|^2}. \tag{8}$$

Proof. Since complex space is also a measurable geometry space, therefore the whole proof is similar to [11, pp. 881–884]. The main difference is that we only need to migrate the inner product in the whole proof from the real domain to complex domain.

Since σ is a complex continuous discriminatory or complex bounded nonlinear piecewise continuous function, $g_i(\mathbf{z})=\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)\in L^2(Z)$ and $\|g_n\|=\int_Z g_n\cdot\overline{g_n}\,d\mathbf{z}\neq 0$. The target function f is continuous, we have $f\in L^2(Z)$. According to Lemma 2.2, $e_n=f-f_n\in L^2(Z)$. Let $\Delta=\|e_{n-1}\|^2-\|e_n\|^2$, then we have

$$\begin{aligned}\Delta&=\|e_{n-1}\|^2-\|e_n\|^2\\&=\langle e_{n-1},e_{n-1}\rangle-\langle e_{n-1}-\beta_n g_n,e_{n-1}-\beta_n g_n\rangle\\&=\langle e_{n-1},e_{n-1}\rangle-(\langle e_{n-1},e_{n-1}\rangle-\langle e_{n-1},\beta_n g_n\rangle\\&\quad-\langle\beta_n g_n,e_{n-1}\rangle+\langle\beta_n g_n,\beta_n g_n\rangle)\\&=\overline{\beta_n}\langle e_{n-1},g_n\rangle+\beta_n\langle g_n,e_{n-1}\rangle-\beta_n\overline{\beta_n}\langle g_n,g_n\rangle\\&=\overline{\beta_n}\langle e_{n-1},g_n\rangle+\beta_n\overline{\langle e_{n-1},g_n\rangle}-\beta_n\overline{\beta_n}\langle g_n,g_n\rangle\\&=\|g_n\|^2\left(\frac{\langle e_{n-1},g_n\rangle\overline{\langle e_{n-1},g_n\rangle}}{\|g_n\|^4}-\left(\beta_n-\frac{\langle e_{n-1},g_n\rangle}{\|g_n\|^2}\right)\right.\\&\quad\left.\times\left(\overline{\beta_n-\frac{\langle e_{n-1},g_n\rangle}{\|g_n\|^2}}\right)\right).\end{aligned} \tag{9}$$

Δ is maximized iff $\beta_n=\langle e_{n-1},g_n\rangle/\|g_n\|^2$, meaning that $\|e_n\|=\|f-(f_{n-1}+\beta_n g_n)\|$ achieves its minimum iff $\beta=\beta_n=\langle e_{n-1},g_n\rangle/\|g_n\|^2$. The result is consistent with the real domain case.

With Lemmas 2.4 and 2.6 we can prove $\|e_n\|$ converges to zero in the same proof method given in [11, pp. 881–884]. For the sake of brevity, readers can refer to [11] for details as it does not convey any new idea to repeat the same proof procedure. □

When the network architecture is fixed (with fixed n), from Theorem 3.1 we have

Theorem 3.2. *Given any complex continuous discriminatory or any complex bounded nonlinear piecewise continuous function $\sigma: C\to C$, for any continuous target function $f:\mathbf{C}^d\to C$ and any function sequence $\{g_n=\prod_{l=1}^{s_n}\sigma(\mathbf{a}_{nl}\cdot\mathbf{z}+b_n)\}$ randomly generated based on any continuous sampling distribution probability, $\lim_{n\to\infty}\|f-f_n\|=0$ holds with probability one if the output parameters are determined by ordinary least square to minimize $\|f(\mathbf{z})-\sum_{i=1}^{n}\beta_i g_i(\mathbf{z})\|$.*

Remark 2. Theorems 3.1 and 3.2 specifies a generic network model where the hidden node itself may be a single-hidden layer feedforward network (SLFN) with multiplicative hidden nodes. For instance, the ith hidden node of this genetic network model is a SLFN with multiplicative hidden node and with activation function σ: $g_i(\mathbf{z})=\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)$. When $s_i=1$, $f_n(\mathbf{z})=$

$\sum_{i=1}^{n}\beta_i\prod_{l=1}^{s_i}\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)=\sum_{i=1}^{n}\beta_i\sigma(\mathbf{a}_{il}\cdot\mathbf{z}+b_i)$ which is a standard SLFNs with one additive hidden layer. Thus, the SLFNs with one additive hidden layer is a specific case of this generic network model when $s_i=1$.

Remark 3. Li et al. [25] proposed a standard fully complex SLFN $s_i=1$ with randomly generated hidden nodes and fixed network architecture, which we called the fully complex extreme learning machine (C-ELM). According to Theorems 3.1 and 3.2, C-ELM can thus, be extended to a more generic model. Furthermore, according to Theorem 3.2, the fully C-ELM with fixed network architectures, where the output parameters are determined by ordinary least square, is a universal approximator if the fully complex activation function σ is complex continuous discriminatory or complex bounded nonlinear piecewise continuous.

3.2. Algorithmic implementation

In this subsection, we introduce an implementation of Theorem 3.1 that resembles the incremental algorithm (I-ELM) [11]. The difference is that hidden node used in this extension may be a multiplicative SLFNs instead of a single additive node and the activation function has been extended from real to complex domain. According to Theorem 3.1, we know that the output weights β_n should be chosen as $\langle e_{n-1},g_n\rangle/\|g_n\|^2$ for newly added hidden node. By the definition of Hermitian inner product, we have $\langle u,v\rangle=\int_X u(\mathbf{z})\overline{v(\mathbf{z})}\,\mathrm{d}\mathbf{z}=\sum_{p=1}^{N}u(\mathbf{z}_p)\overline{v(\mathbf{z}_p)}$, thus an estimate β_n based on the training samples is

$$\beta_n=\frac{E_{n-1}\cdot H^*}{H\cdot H^*}=\frac{\sum_{p=1}^{N}e_{n-1}(p)\overline{g_n(p)}}{\sum_{p=1}^{N}g_n(p)\overline{g_n(p)}},\tag{10}$$

where H^* means complex conjugate transposition, $g_n(p)$ is the output of the nth hidden node in the complex network for the input of pth training sample and $e(p)$ is the corresponding residual error before this new hidden node is added. $H=[g_n(1),\ldots,g_n(N)]^{\mathrm{T}}$ is the activation vector of the new node for all the N training samples and $E_{n-1}=[e_{n-1}(1),\ldots,e_{n-1}(N)]^{\mathrm{T}}$ is the residual vector before adding the new hidden node. In real applications, one may not really wish to get zero approximation error by adding infinite number of nodes to the network by providing a maximum number of hidden nodes. The detail algorithm is summarized as follows:

Algorithm. Given a training set $\aleph=\{(\mathbf{z}_i,t_i)|\mathbf{z}_i\in\mathbf{C}^d,t_i\in C,i=1,\ldots,N\}$, complex activation function σ, maximum number of hidden nodes $L_{\max}$ and expected learning accuracy ε,

Step 1: **Initialization:** Let $L=0$ and residual error $E=t$, where $t=[t_1,\ldots,t_N]^{\mathrm{T}}$.

Step 2: **Learning step:**

while $L<L_{\max}$ and $\|E\|>\varepsilon$

(a) Increase the number of hidden nodes L: $L=L+1$.

(b) Assign randomly the hidden node parameters $(\mathbf{a}_L,b_L)$ for new hidden node L.

(c) Calculate the output weight β_L for the new hidden L:

$$\beta_L=\frac{E\cdot H_L^*}{H_L\cdot H_L^*}.\tag{11}$$

(d) Calculate the residual error after adding the new hidden node L:

$$E=E-\beta_L\cdot H_L.\tag{12}$$

endwhile

4. Experimental verification

In the previous section, we have provided our theoretical justification for the incremental feedforward networks in the complex domain. In this section, simulation results are given to verify the theory.

For the sake of simplicity, we demonstrate the universal approximation capability of complex I-ELM with one additive hidden layer ($s_i=1$) and with three fully complex activation functions: $\arcsin(z)=\int_0^z \mathrm{d}t/(1-t)^{1/2}$, $\arccos(z)=\int_0^z \mathrm{d}t/(1-t^2)^{1/2}$, and $\mathrm{arcsinh}(z)=\int_0^z \mathrm{d}t/(1+t^2)^{1/2}$, where $z\in C$. All simulations were conducted in MATLAB environment running in a P4/2.8 GHz PC.

4.1. Function approximation

Two approximation problems in complex domain used in [1] have been investigated first. In these simulations, 10 000 training samples and 1000 testing samples are randomly drawn from the interval $[0+\mathrm{i}0,1+\mathrm{i}]$. Both the input weight vectors $\mathbf{a}_i$ and biases b_i of the complex I-ELM are randomly chosen from a complex area centered at the origin with the radius set to 1. $L_{\max}$ is set to 6000 and $\varepsilon=0.01$. The simulation results are obtained after 10 independent runs for each case.

Example 1. The convergence of I-ELM with the fully complex activation functions is first verified in a non-analytic function used in [1]:

$$f(z)=f(x+\mathrm{i}y)=\mathrm{e}^{\mathrm{i}y}(1-x^2-y^2).\tag{13}$$

Fig. 1 shows the update of the average testing root mean square error (RMSE) when the network grows. It can be seen that the learning convergence curves decrease with the increase of network size. Fig. 2 shows that the training time is linearly increasing with the increase of network size, which is consistent with the analysis on real domain.

Example 2. The convergence of I-ELM with the fully complex activation functions is also verified through an analytic function given as

$$f(z)=f(x+\mathrm{i}y)=\sin(x)\cosh(y)+\mathrm{i}\cos(x)\sinh(y).\tag{14}$$

Figs. 3 and 4 show the update of the average testing RMSE and the spent time with the increase of hidden nodes, respectively. We observe that the learning convergence

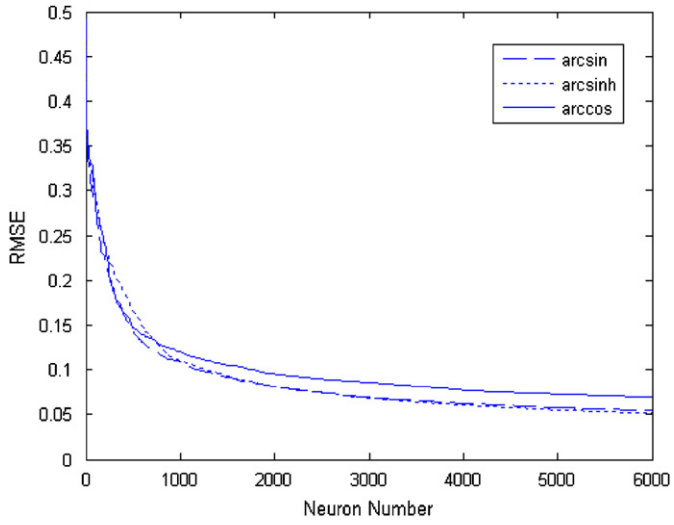


Fig. 1. Learning convergence for arcsin, arcsinh and arccos activation functions (Example 1).

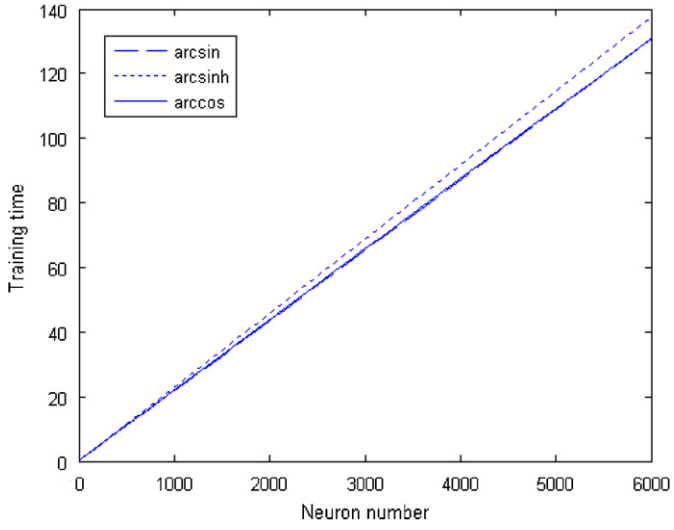


Fig. 2. Training time for different activation functions (Example 1).

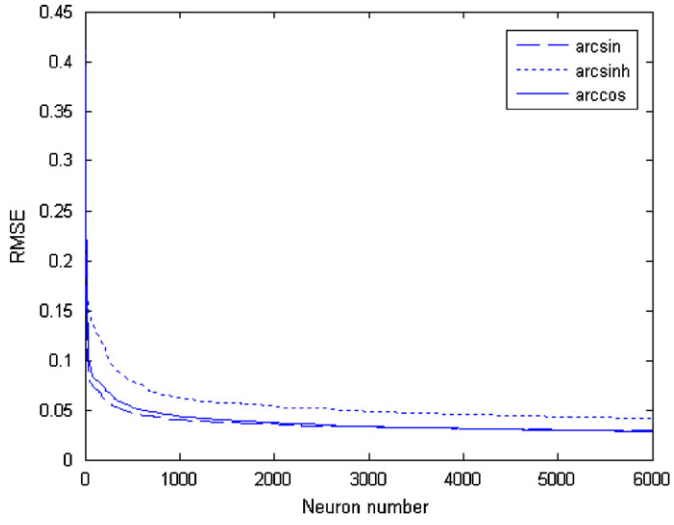


Fig. 3. Learning convergence for arcsin, arcsinh and arccos activation functions (Example 2).

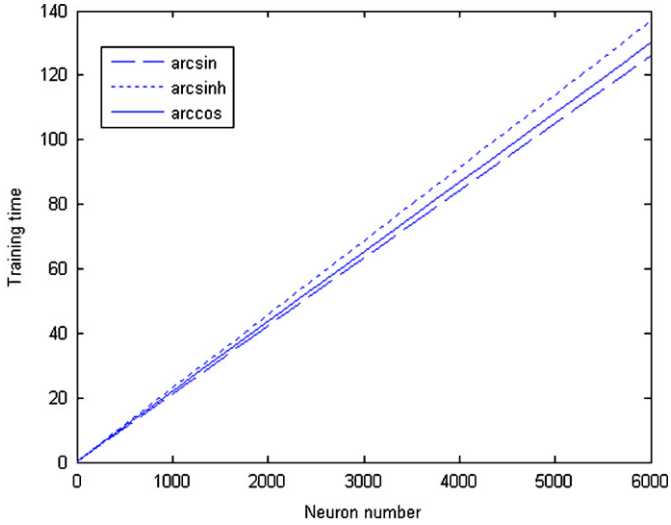


Fig. 4. Training time for different activation functions (Example 2).

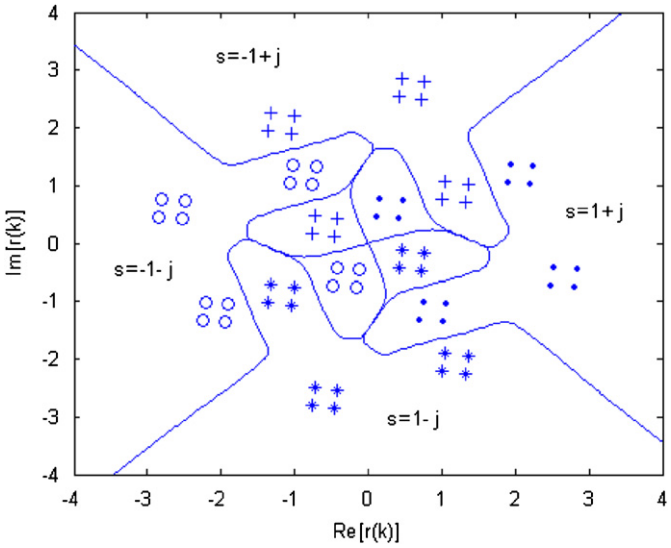


Fig. 5. Decision boundary of Bayesian equalizer.

curves decrease and the training time is linearly increasing with the increase of network size.

4.2. Channel equalization

It is well known that the channel equalization can be considered as a classification problem. Here, a third-order complex channel model studied by Chen et al. [8] for 4-QAM signaling is used to verify the performance of I-ELM with the fully complex activation functions. The channel model is given by

$$A(z) = (0.7409 - \mathrm{j}0.7406)(1 - (0.2 - \mathrm{j}0.1)z^{-1}) \times (1 - (0.6 - \mathrm{j}0.3)z^{-1}). \quad (15)$$

The noise variance is $\sigma_e^2 = 0.06324$ (SNR = 15 dB). Similar to Chen et al. [8], the equalizer dimension was set to $m = 1$ and the equalizer decision delay was $\tau = 0$. 8000 training samples are used to train the complex I-ELM equalizer and the maximum hidden node number is set to 5000. The real and imaginary part of complex input layer weights and biases are randomly chosen from the interval $[-1, 1]$. Fig. 5 shows the decision boundary using Bayesian equalizer which can achieve the optimal solution. The symbols $(\bullet, *, +, o)$ represent the four classes of input signals. The complex I-ELM equalizers with different activation functions are shown in Figs. 6–8. It can be seen that complex I-ELM equalizers can also separate the input space into four areas clearly.

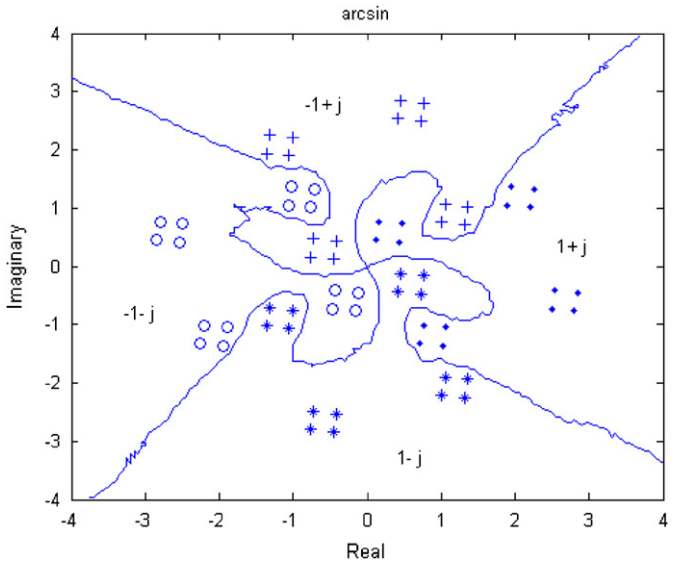


Fig. 6. Decision boundary of complex I-ELM equalizer with arcsin activation function.

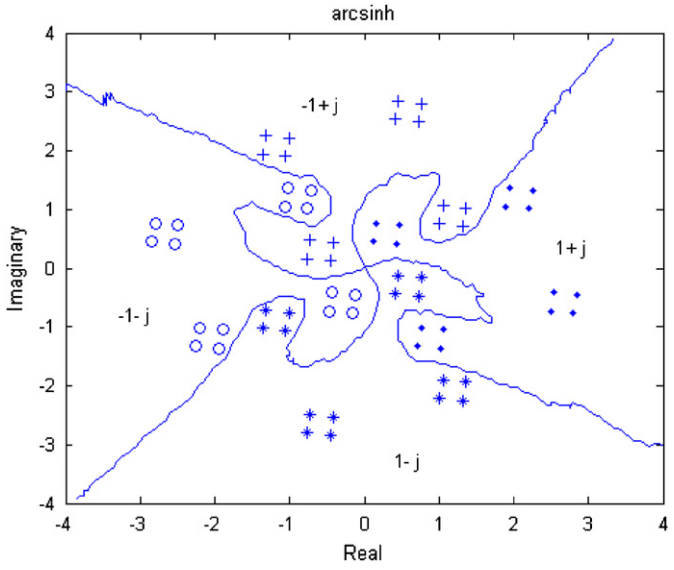


Fig. 7. Decision boundary of complex I-ELM equalizer with arcsinh activation function.

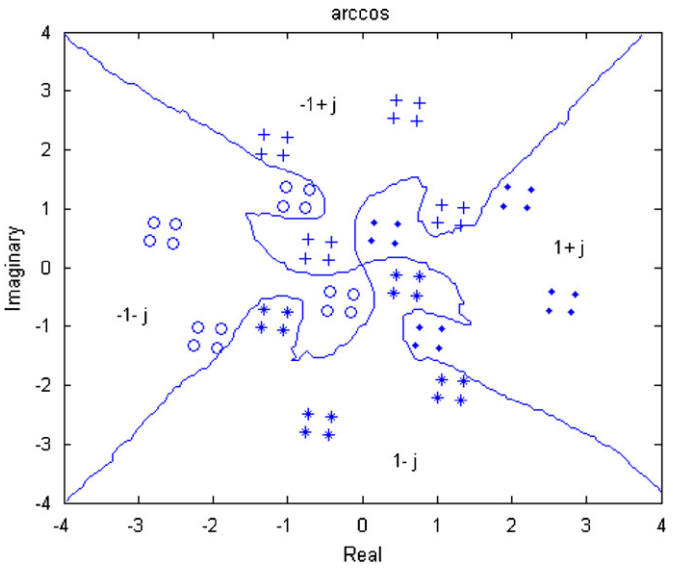


Fig. 8. Decision boundary of complex I-ELM equalizer with arccos activation function.

5. Conclusions

In this paper, we show that the complex SLFNs using the proposed incremental algorithm (I-ELM) can approximate any target continuous functions in complex domain. Each hidden node in I-ELM can be a single additive node or a multiplicative combination of additive nodes. In contrast to tuning-based learning algorithms, our tuning-free I-ELM does not requires any intervention from users. The proposed I-ELM can be applied to a wide range of complex activation functions which may be differentiable or non-differentiable. As long as the hidden layer activation function is complex continuous discriminatory or complex bounded nonlinear piecewise continuous I-ELM can still approximate any target functions in the complex domain. The traditional gradient descent based learning algorithms cannot be applied to networks with non-differential activation functions and are limited by local minima issues. However, the proposed I-ELM can avoid the above issues.

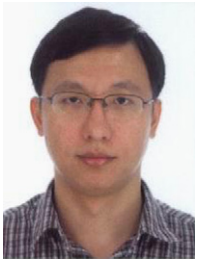
References

[1] P. Arena, L. Fortuna, R. Re, M.G. Xibilia, Multilayer perceptrons to approximate complex valued functions, Int. J. Neural Syst. 6 (4) (1995) 435–446.
[3] A.R. Barron, Universal approximation bounds for superpositions of a sigmoid function, IEEE Trans. Inf. Theory 39 (3) (1993) 930–945.
[4] E. Baum, On the capabilities of multilayer perceptrons, J. Complexity 4 (1988) 193–215.
[5] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.
[6] I. Cha, S.A. Kassam, Channel equalization using adaptive complex radial basis function networks, IEEE J. Sel. Areas Commun. 13 (1) (1995) 122–131.

[7] S. Chen, S. Mclaughlin, B. Mulgrew, Complex-valued radial basis function networks, part i: network architecture and learning algorithms, Signal Process. 35 (1) (1994) 19–31.
[8] S. Chen, S. Mclaughlin, B. Mulgrew, Complex-valued radial basis function networks, part II: application to digital communications channel equalization, Signal Process. 36 (1994) 175–188.
[9] S. Ferrari, R.F. Stengel, Smooth function approximation using neural networks, IEEE Trans. Neural Networks 16 (1) (2005) 24–38.
[10] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3072.
[11] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.
[12] G.-B. Huang, C.-K. Siew, Extreme learning machine: RBF network case, in: Proceedings of the Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004), Kunming, China, Avon Books, New York, 6–9 December, 2004, pp. 1029–1036.
[13] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, Int. J. Inf. Technol. 11 (1) (2005).
[14] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of International Joint Conference on Neural Networks (IJCNN2004), Budapest, Hungary, 25–29 July, 2004, pp. 985–990.
[15] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Real-time learning capability of neural networks, IEEE Trans. Neural Networks 17 (4) (2006) 863–878.
[16] G.-B. Huang, Q.-Y. Zhu, K.Z. Mao, C.-K. Siew, P. Saratchandran, N. Sundararajan, Can threshold networks be trained directly?, IEEE Trans. Circuits Syst. II 53 (3) (2006) 187–191.
[17] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[18] B. Igelnik, Y.-H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. on Neural Networks 6 (6) (1995) 1320–1329.
[19] D. Jianping, N. Sundararajan, P. Saratchandran, Communication channel equalization using complex-valued minimal radial basis function neural networks, IEEE Trans. Neural Networks 13 (3) (2002) 687–696.
[20] T. Kim, T. Adali, Fully complex backpropagation for constant envelope signal processing, in: Proceedings of the 2000 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing X, vol. 1, 2000, pp. 231–240.
[21] T. Kim, T. Adali, Universal approximation of fully complex feedforward neural networks, in: International Conference on Acoustics, Speech, and Signal Processing, vol. 1, 2002, pp. I-973–I-976.
[22] T. Kim, T. Adali, Approximation by fully complex multilayer perceptrons, Neural Comput. 15 (2003) 1641–1666.
[23] A.N. Kolmogorov, S.V. Fomin, Elements of the Theory of Functions and Functional Analysis, Volume 2: Measure, The Lebesgue Integral, Hilbert Space, Graylock Press, Baltimore, MD, 1961.
[24] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks 6 (1993) 861–867.
[25] M.-B. Li, G.-B. Huang, P. Saratchandran, N. Sundararajan, Fully complex extreme learning machine, Neurocomputing 68 (2005) 306–314.
[26] D. Lowe, Adaptive radial basis function nonlinearities and the problem of generalisation, in: Proceedings of First IEE International Conference on Artificial Neural Networks, 1989, pp. 171–175.
[27] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (1991) 246–257.
[28] S. Tamura, M. Tateishi, Capabilities of a four-layer feedforward neural networks: four layers versus three, IEEE Trans. Neural Networks 8 (2) (1997) 251–255.

Guang-Bin Huang received his B.Sc. degree in applied mathematics and M.Eng. degree in computer engineering from Northeastern University, PR China, in 1991 and 1994, respectively, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 1999. During his undergraduate period, he also concurrently studied in Wireless Communication, Department of Northeastern University, PR China.

From June 1998 to May 2001, he worked as Research Fellow in Singapore Institute of Manufacturing Technology (formerly known as Gintic Institute of Manufacturing Technology) where he has led/implemented several key industrial projects. From May 2001, he has been working as an Assistant Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University. His current research interests include extreme learning machine, machine learning, bioinformatics and networking. He is an associate editor of *IEEE Transactions on Systems, Man and Cybernetics—Part B and Neurocomputing. He is a senior member of the IEEE.*



Lei Chen received his B.Sc. degree in applied mathematics and his M.Sc. degree in operational research and control theory from Northeastern University, PR China, in 1999 and 2002, respectively, and his Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2007. Now he is a postdoctoral fellow in National University of Singapore, Singapore. His research interests include artificial neural networks, pattern recognition and machine learning.

Ming-Bin Li was born in Liaoning, China, in 1975. He received his B.Eng. degree from the Shenyang Institute of Technology, China in 1998 and his M.Eng. degree from Northeastern University, China, in 2001. He obtained his Ph.D. degree in Nanyang Technological University (NTU), Singapore in 2006. He is currently a research associate at Intelligent Systems Centre, Nanyang Technological University.

His main research interests include neural network, fuzzy logic, system modeling, channel equalization and dynamic control.

Chee-Kheong Siew obtained his B.Eng., M.Sc. and Ph.D. from University of Singapore, Imperial College, UK and NTU, Singapore, respectively. He is currently an Associate Professor in the School of EEE, Nanyang Technological University (NTU), Singapore. From 1995 to 2005, he served as the Head of Information Communication Institute of Singapore after he managed the transfer of ICIS to NTU and rebuilt the institute in the university environment. After 6 years in the industry, he joined NTU in 1986 and was appointed as the Head of the Institute in 1996. He has served in various conference technical program committees and also as reviewer for various journals. His current research interests include neural networks, packet scheduling, traffic shaping, admission control, service curves and admission control, QoS framework, congestion control, multipath routing and intelligent networks. He is a member of IEEE.